Memories

Chapter 11

Virtual Memory Technologies And Virtual Addressing

Topics

- Introduction
- Definition of Virtual Memory
- Virtual Memory Terminology
- Interface To Multiple Physical Memory Systems
- Example of a Virtual Memory System
- Address Translation Or Address Mapping
- Avoiding Arithmetic Calculation
- Discontinuous Address Spaces
- Other Memory Organizations

Topics

- Discontinuous Address Spaces
- Other Memory Organizations
- Motivation for Virtual Memory
- Multiple Virtual Spaces and Multiprogramming
- Technologies to Create Dynamic Virtual Memories
- Base-Bound Registers
- Changing the Virtual Space
- Protection
- Segmentation

Topics

- Demand Paging
- Page Fault and Replacement
- Terminology
- Address Translation
- Using Powers Of Two
- Control Bits
- Storing Page Tables
- Translation Look-aside Buffer
- Consequences For Programmers
- Summary

Introduction

This chapter covers

- Motivation
- Technologies to create virtual address spaces
- Mapping between virtual and physical memory
- How operating system (OS) uses virtual memory

Definition of Virtual Memory

- A mechanism that hides details of underlying physical memory to provide more convenient memory environment
- A memory access scheme to overcome limitations of physical memory and addressing scheme
- An example from previous chapter
  - A controller that provides byte address while underlying physical memory uses word addressing choosing powers of two avoids arithmetic computation and makes translations of byte addresses to word address trivial

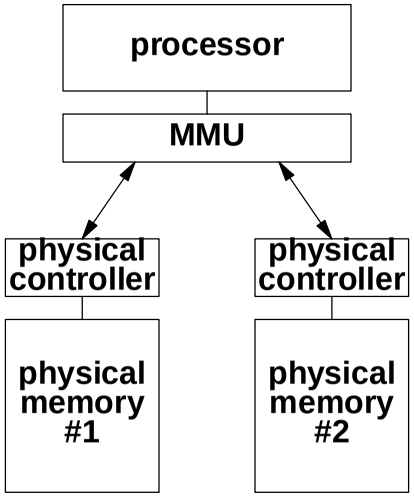
Virtual Memory Terminology

- Memory Management Unit (MMU)
  - An intelligent memory controller that creates virtual address space
  - Processor generates virtual addresses

- Real: a synonym for physical, i.e.
  - real memory = physical memory
  - real address = physical address
  - real address space = physical address space

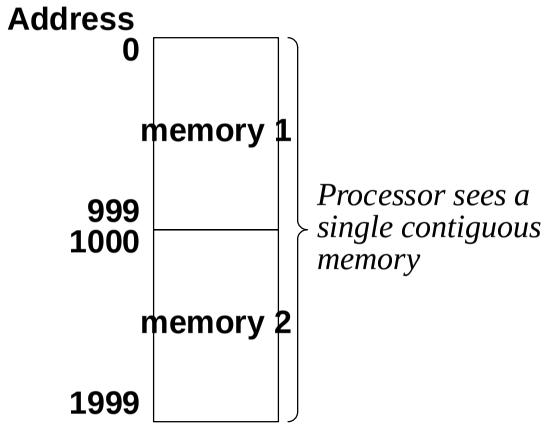
Interface To Multiple Physical Memory Systems

- Example of connecting dissimilar memories
  - Figure shows a 4 byte/word SRAM and 8 byte/word DRAM connected together
  - Both physical memories are integrated into a single virtual address space



Architecture in which two dissimilar memories connect to processor.

Example of a Virtual Memory System



Virtual memory system that divides an address space among two physical memories. MMU uses address to decide which memory to access.

Address Translation Or Address Mapping

```
receive memory request from processor;
let A be the address in the request;
if (A > 1000) {
        A = A - 1000;
        pass the modified request to memory 2;
} else {
        pass the unmodified request to memory 1;
}
```

Algorithm used by MMU to create virtual memory shown in previous figure. The MMU maps virtual address space onto two physical memories.

Address Translation Or Address Mapping

A MMU can use multiple underlying physical memory systems to provide a processor with the illusion of a single, uniform memory system. Because each underlying memory uses address that start at zero, the MMU must translate between the addresses used by the processor and the addresses used by each memory

Avoiding Arithmetic Calculation

- A computation such as subtraction involves both hardware and time
- Solution
  - Divide address space along boundaries that correspond to powers of two
  - Now MMU looks at specifi c bit(s) of the virtual address to decide which memory to access

    Dividing a virtual address space on a boundary that corresponds to a power of two allows the MMU to choose a physical memory and perform the necessary address translation without requiring arithmetic operations.

Addresses	Values In Binary
0	0 0 0 0 0 0 0 0 0 0 0
to	to
1023	0 1 1 1 1 1 1 1 1 1 1
1024	1 0 0 0 0 0 0 0 0 0 0
to	to
2047	1 1 1 1 1 1 1 1 1 1 1

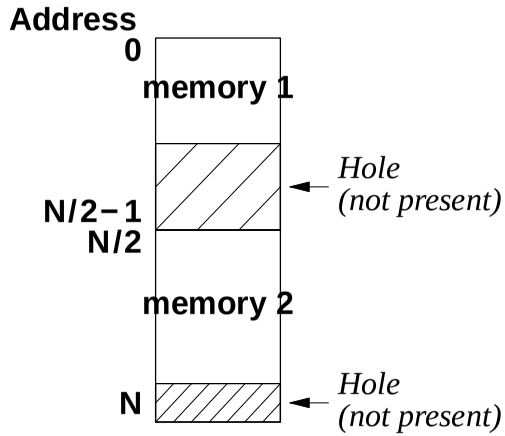
Binary values of addresses in the range 0 to 2047. Note, addresses above 1023 are identical except for the highest order bit.

Discontinuous Address Spaces

- What happens if arbitrary amount of memory is installed?
- Example, M1 = 1024 and M2 = 812 bytes in the previous example.
  - each physical memory address has a corresponding virtual address
  - however, some virtual addresses do not correspond to any physical memory
  - virtual addresses 1836-2047 do not correspond to any physical memory
- These are called *holes*.

Discontinuous Address Spaces

A virtual address space can be contiguous, in which case every address maps to a location of an underlying physical memory, or non contiguous, in which case the address space can contain holes. If a processor attempts to read or write any address that does not correspond to physical memory, an error results.



Virtual address space of N bytes mapped into two physical memories. The space is not contiguous because only part of each memory is present.

Other Memory Organizations

- Interleaving bytes among different physical memory modules
  - Example: 4 bytes of a word interleaved among 4 physical memories
  - Low-order 2 bits specifi es the four bytes
  - Advantage: can access the four bytes simultaneously

Motivation for Virtual Memory

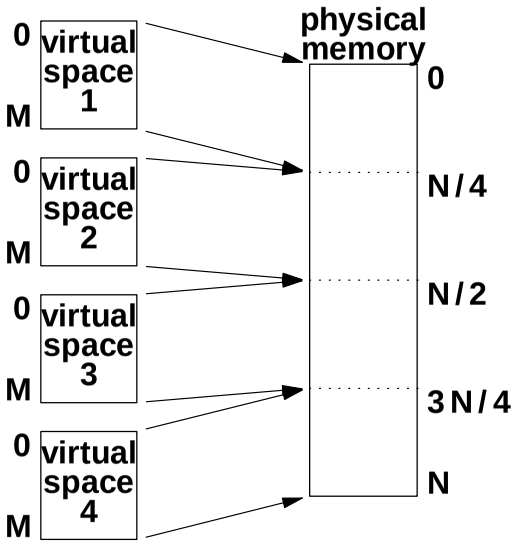
- Homogeneous integration of hardware
  - different size, types, word-size, physical memories can be integrated under single virtual space (see earlier example)
- Programming convenience
  - without virtual addresses, processor will need separate addresses for each physical memory.
  - Programming and changes to memory access would be diffi cult for programmer

Motivation for Virtual Memory

- Support for multiprogramming
  - helps in allowing multiple programs to run at the same time
- Protection of programs and data
  - CPU uses modes of execution to determine which instruction is allowed at any time
  - virtual memory linked to protection

Multiple Virtual Spaces and Multiprogramming

- How to avoid using same memory location by two or more programs running at the same time?
- Use virtual memory to establish a separate virtual address space for each program
- Old method: partitioning, split physical memory into areas for each program
  - advantage, no conflict
  - disadvantage, total area available for a program is a fraction of physical memory



Virtual address spaces mapped onto a single physical memory.

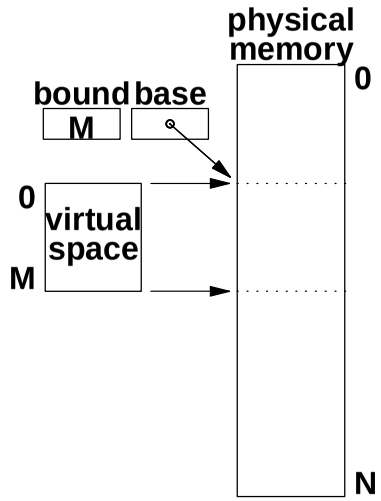
Technologies to Create Dynamic Virtual Memories

- Base-bound registers
- Segmentation
- Demand Paging

Base-Bound Registers

- Creates a single virtual address space and maps the space onto a region of physical memory
- Base and bound are two registers and a part of MMU
  - base register species location of virtual space
  - bound register specifi es the size of address space.

Changing the Virtual Space



Virtual memory that uses base-bound mechanism.

A base-bound mechanism uses two values in the MMU to specify how the virtual address space maps onto the physical address space. The base-bound mechanism is powerful because an operating system can change the mapping dynamically.

Protection

- A base register can map from virtual to physical address
- This does not prevent a program from accidently of maliciously referencing large memory locations
- The bound register is used to ensure that program will not exceed allocated space.
- The MMU checks memory reference to bound register values and raises an error if address is too large.

Segmentation

Segmentation refers to a virtual memory scheme in which programs are divided into variable-size blocks, and only the blocks currently needed are kept in memory. Because it leads to a problem known as memory fragmentation, segmentation is seldom used

Demand Paging

- An alternative to segmentation.
- Scheme is as follows
  - divide program into pieces
  - keep it in external storage till needed
  - load when needed
- Pieces are fi xed-size blocks called pages. Size of pages used to be 512 bytes or 1 KB, nowdays Pentium 4KB
- Hardware handles address mapping and detects missing pages
- Software moves pages between external store and physical memory

Page Fault and Replacement

- Page Fault: when a reference is made to a missing page
- Page replacement
  - Choosing page to be removed from physical memory, when it is full and new page needs to be loaded.
  - Software algorithms handle page replacement
  - Pages not referenced for a while could be considered for replacement (note: not true for all algorithms)

Terminology

- Page
  - block of program address space
  - unit of memory moved at a time from external storage to physical memory
- Frame
  - slot of physical memory that can hold a page
- Resident
  - page in memory

Terminology

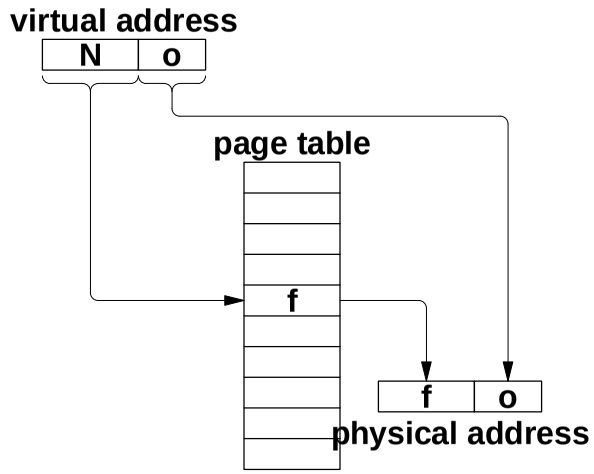
- Page table
  - a 1D list indexed by page number. Value is null if page is not resident, and corresponding frame number for pages which are resident in physical memory.

Address Translation

- Assume each page is K bytes
- Example: translating virtual address V, to corresponding physical address, P.
  - determine number of page where address V lies
    N = int (V/K)
  - use page number as index into page table to fi nd corresponding frame number in memory
  - determine offset within the page
    O = C mod K
    P = pagetable[N] + O

Using Powers Of Two

- Using powers of two eliminates the need for devision and remainder computation.



MMU performing address translation in a paging system

Control Bits

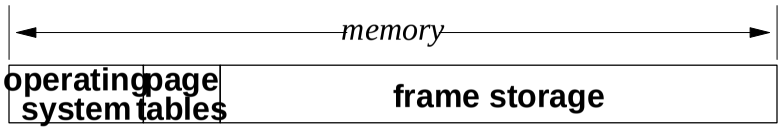
- Presence bit
  - whether page is currently in memory
  - helps detect page fault
- Use bit
  - provides information for page replacement
  - set whenever page is accessed.
  - page not referenced recently is candidate for eviction
- Modify bit
  - set whenever a page is written after it was loaded.
  - used during page replacement whether page needs to be written back to external storage

Storing Page Tables

- Page tables are accessed frequently,and determine performance of memory access.
- Page tables can also be quite large.

- Where are page tables stored? Possible locations
  - Special MMU chip external to processor, with high speed hardware interface between processor-MMU
  - In the physical memory. SRAM for page table, DRAM for frames
  - Subset of page table (TLB) may be stored in high speed memories like CAM

Storing Page Tables



A part of the physical memory reserved to store page tables.

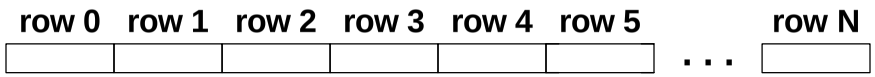
Translation Look-aside Buffer

- High speed hardware to optimize performance of demand paging
- Provides fast table lookups, TLB is a CAM like storage device.
- Processor stores page table in memory and subset of page table in TLB. Simultaneously accesses both memory and TLB for page translation. Accepts first reply.
- Processor tends to fetch successive instructions from same page. Recently accessed page table entries are stored in TLB.

Consequences For Programmers

- Programmers can affect virtual memory performance.
- Example 2 D array stored in memory.
  - Can be stored as row-major or column-major
  - All elements of a row stored in contiguous memory locations
  - Programmer writing nested loop for A[i,j] with row-major storage should have inner loop as j, and inner loop i with column-major storage.
  - Ensures contiguous memory are accessed in sequence, all in the same page
  - Results in less page faults

Consequences For Programmers



2D array stored in row-major order. A row is contiguous in memory.

Summary

- Virtual memory system hides details of underlying physical memory
- Virtual memory is convenient for programming, and supports multiprogramming and protection
- Base-bound, Segmentation, and Demand Paging are three virtual technologies
- Demand paging, the most popular among the three uses a page table to map virtual address to physical address
- Important subset of page table is stored in TLB, a high speed device.
- Using powers of two helps avoid arithmetic operations